

(19) 日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-267654

(P 2000-267654 A)

(43) 公開日：平成12年9月29日 (2000. 9. 29)

(51) Int. Cl.

G10H 1/00

識別記号

102

1/40

G10L 15/10

H04R 1/00

310

F 10H 1/00 B 5D015 (参考)

G10H 1/00 B 5D020

1/40 B 5D378

G10L 15/00 B 531 (参考)

H04R 1/00 E 310 (参考)

審査請求 未請求 請求項の数 4 O L (全13頁) 最終頁に続く

(21) 出願番号

特願平11-72390

(71) 出願人 000000491

アイワ株式会社 東京都台東区池之端1丁目2番11号

(22) 出願日

平成11年3月17日 (1999. 3. 17)

(72) 発明者 植木 亨 東京都台東区池之端1丁目2番11号 アイ

ワ株式会社内

(72) 発明者 秋本 義雄 東京都台東区池之端1丁目2番11号 アイ

ワ株式会社内

(74) 代理人 100079049 井理士 (外3名)

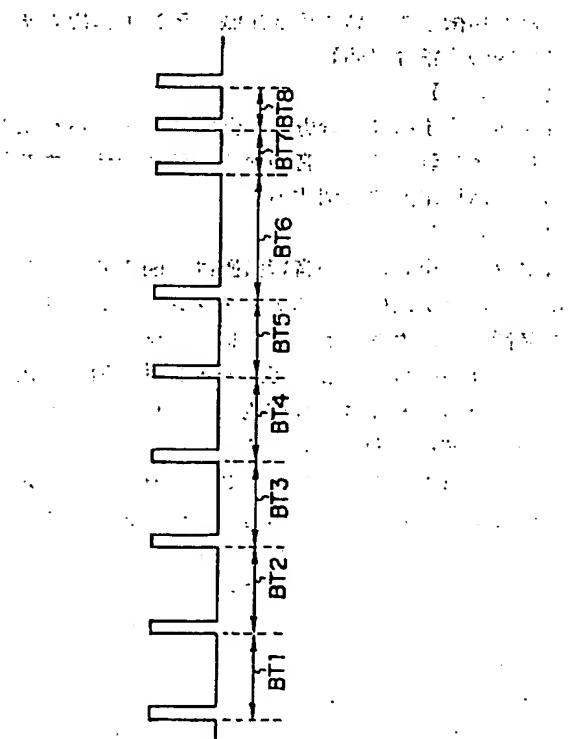
井理士 (外3名)

F ターム (参考) 5D015, AA06, CC13

5D020, AC00, 5D378, AD62, MM14,

(54) 【発明の名称】テンポ算出方法

(57) 【要約】(1) 本発明は、音楽のテンポを算出する方法である。
【課題】(2) テンポ算出時間を短縮する。
【解決手段】(3) ビートを連続して検出し、連続して検出したビートの間隔時間であるビート時間と連続して検出したビート時間に近い過去のビート時間の個数を算出する。過去のM個のビート時間 (BT1 ~ BT8) 中で、今回のビート時間に近いビート時間の個数を算出する。今回のビート時間に近い過去のビート時間の個数がN以上の場合には、今回のビート時間と今回のビート時間と近い過去のビート時間との平均ビート時間を算出する。テンポとして、60を平均ビート時間で割った値に設定する。以上の処理により、テンポを算出することができると共に、算出したテンポを、音楽の実際にテンポに変更(近づける)することができる。



【特許請求の範囲】

【請求項1】 音源からの音楽から、該音楽のテンポに沿って周期的に発生する周期音を連続して検出し、

前記連続して検出された周期音に基づいて、連続する2つの周期音間隔時間である周期音時間を連続して求め、過去求めたM（Mは2以上の自然数）個の周期音時間の中に今回求めた周期音時間に近い周期音時間の個数を算出し、

算出した個数が前記M個の内のN（NはMより小さい自然数）個以上の場合、今回求めた周期音時間及び今回求めた周期音時間に近い過去の周期音時間から、平均周期音時間を求め、

求めた平均周期音時間からテンポを算出する、

テンポ算出方法。

【請求項2】 過去求めたM個の周期音時間の中に、連続する2つの周期音時間の一方が他方の3倍である組み合わせが、所定組数ある場合には、該2つの周期音時間の平均時間を、前記平均周期音時間とすることを特徴とする請求項1記載のテンポ算出方法。

【請求項3】 前記周期音のレベルの立ち上がりを検出することにより、周期音を検出する請求項1又は請求項2記載のテンポ算出方法。

【請求項4】 前記周期音をサンプリングし、今回サンプリングした周期音が前回周期音検出時から所定時間経過して検出されたものであると共に今回サンプリングした周期音のレベルがしきい値以上でありかつ過去サンプリングされた所定個数の周期音のレベルより大きい場合に、前記周期音のレベルの立ち上がりを検出する、

ことを特徴とする請求項3記載のテンポ算出方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、テンポ算出方法に係り、より詳しくは、音源からの音楽のテンポを算出するテンポ算出方法に関する。

【0002】

【従来の技術】 従来の音響装置は、図14に示すように、音源（例えば、ミニディスクレコーダ14、CD再生装置16、カセットテープレコーダ18、チューナー20）からの音のうち、音楽のテンポに沿って周期的に発生する音、例えば、高域側（3kHz～15kHz）の所定周波数（例えば3.4kHz）の音（アタック音）、低域側（100Hz～300Hz）の所定周波数

（例えば、105Hz）の音（ビート音）を検出するスペクトラムアナライザ1C28を備えている。スペクトラムアナライザ1C28はマイクロコンピュータ（以下、マイコン26という）及びリズム専用マイコン35に接続されている。マイコン26には、操作部24及び表示部30が接続されている。リズム専用マイコン35には、複数の楽器音を組み合わせて構成される複数のリ

ズムパターンを選択的に出力する音声出力IC32が接続されている。音声出力IC32及び音源はアンプ22に接続されている。

【0003】 そして、従来の音響装置は、表示部30を制御するマイコン28の他に、スペクトラムアナライザ1C28からの出力に基づいて、音源からの音楽にリズムパターンをミックスして出力するように、音声出力ICを双方向通信により制御する専用のリズム専用マイコン35を備えている。即ち、リズム専用マイコン35は、音源からの音楽のテンポを自動的に算出する。

【0004】 このような音響装置では、音楽のテンポは種々のパターンがあるため、各パターンごとにマッチングをとり、最も近いパターンを選択することにより、テンポを求めている。

【0005】

【発明が解決しようとする課題】 しかし、上記音響装置は、各パターンごとにマッチングをとり、最も近いパターンを選択することにより、テンポを求めているので、正確ではあるが、テンポ算出するまでに長時間を要する。本発明は上記事実に考えなされたもので、テンポ算出時間を短縮させることの可能なテンポ算出方法を提案することを目的とする。

【0006】

【課題を解決するための手段】 上記目的達成のため請求項1記載の発明は、音源からの音楽から、該音楽のテンポに沿って周期的に発生する周期音を連続して検出し、前記連続して検出された周期音に基づいて、連続する2つの周期音間隔時間である周期音時間を連続して求め、過去求めたM（Mは2以上の自然数）個の周期音時間の中に今回求めた周期音時間に近い周期音時間の個数を算出し、算出した個数が前記M個の内のN（NはMより小さい自然数）個以上の場合、今回求めた周期音時間及び今回求めた周期音時間に近い過去の周期音時間から、平均周期音時間を求め、求めた平均周期音時間からテンポを算出する。

【0007】 このように、連続する2つの周期音時間を連続して求め、今回求めた周期音時間と過去求めたM個の周期音時間の中の今回求めた周期音時間に近い周期音時間とから平均周期音時間を求めてテンポを算出する。即ち、複数のパターンをマッチング処理しないで、テンポを算出するので、簡易にテンポを算出することができ、よって、テンポ算出する時間を短くすることができる。

【0008】 ところで、過去求めたM個の周期音時間の中に、連続する2つの周期音時間の一方が他方の3倍である組み合わせが、所定組数ある場合には、該2つの周期音時間の平均時間を、前記平均周期音時間とする。即ち、音楽の中には、周期音が一定間隔に発生するものばかりでなく、連続する2つの周期音時間の一方が他方の3倍である曲があるので、このような曲のテンポも、複

数のパターンをマッチング処理しないで、簡易に求めることができる。

【0009】また、周期音のレベルの立ち上がりを検出することにより、周期音を検出するようにしてもよい。この場合は、周期音をサンプリングし、今回サンプリングした周期音が前回周期音検出時から所定時間経過して検出されたものであると共に今回サンプリングした周期音のレベルがしきい値以上でありかつ過去サンプリングされた所定個数の周期音のレベルより大きい場合に、前記周期音のレベルの立ち上がりを検出するとする。

【0010】

【発明の実施の形態】以下、図面を参照して、本発明の実施の形態を詳細に説明する。

【0011】図1に示すように、音響装置（オーディオシステム）10は一対のスピーカ12、ミニディスクレコーダ14、CD再生装置16、カセットテープレコーダ18、チューナー20、操作部24及び表示部30を備えたアンプ22を備えている。

【0012】図2には、本実施の形態に係る音響装置の主制御部が記載されている。図2に示すように、本制御系は、音源（例えば、ミニディスクレコーダ14、CD再生装置16、カセットテープレコーダ18、チューナー20）からの音のうち、音楽のテンポに沿って周期的に発生する音、例えば、高域側（3 kHz～1.5 kHz）の所定周波数（例えば3.4 kHz）の音（アタック音）、低域側（100 Hz～300 Hz）の所定周波数（例えば1.05 Hz）の音（ビート音）を検出するスペクトラムアナライザIC28を備えている。スペクトラムアナライザIC28はマイクロコンピュータ（以下、マイコン26という）に接続されている。マイコン26には操作部24及び表示部30が接続されている。

更に、マイコン26には、所定の楽器音や図3に示すように、複数の楽器音を組み合わせて構成される複数のリズムパターンを選択的に出力する音声出力IC32が接続されている。音声出力IC32及び音源はアンプ22に接続されている。マイコン26は音源（14～20）を制御する。このように、音源を制御するマイコンが音声出力ICを制御するので、複数の機能を兼用することができ、装置構成を簡略化することができる。即ち、従来の音響装置のように専用のリズム専用マイコンを省略している。

【0013】次に、本実施の形態の作用を説明する。

【0014】本実施の形態では、後述するように、上記音源からの音楽から、ビート音を検出し、検出したビート音から音楽のテンポを算出し、音楽のテンポに対応する1小節の時間内で、ユーザーによって予め選択されたリズムパターンを出力するように音声出力ICを、単方向に制御する。即ち、マイコン26は、音声出力IC32の出力状態にかかわらず、リズムパターンを出力するように命令する。よって、音声出力IC32は、強制的

にリズムパターンを出力する。このようにマイコンが音声出力ICを単方向に制御するので、双方向通信する場合と比較すると、処理を簡易なものにできる。

【0015】以後、音声出力ICは、このマイコン26からの命令に従って、上記テンポに対応する1小節の時間内でリズムパターンを繰り返し出力する。

【0016】ここで、リズムパターンは、図3（A）～図3（D）に代表的に示されるように、各リズムパターンは複数の楽器を音を組み合わせて構成されている。各リズムパターンは、1小節時間に例えればアクセント数（図3（A）、図3（B）は16個、図3（C）、図3（D）は12個）等で割った各分割時間ごとに、各楽器音の出力タイミングが設定されている。図3には代表的なリズムパターンが示されており、リズムパターンはこれらに限定されるものでない。また、これらリズムパターンは、選択手段を設けることにより、ユーザーにより、設定可能とする構成をとることもできる。

【0017】一方、音声出力ICは、上記テンポに対応する1小節の時間内でリズムパターンを繰り返し出力するが、繰り返し出力するリズムパターンが音楽のテンポにずれる場合があるので、本実施の形態では、リズムパターンを音楽のテンポに同期させるリズムパターン同期処理を実行する。

【0018】以下、上記各処理を詳細に説明する。

【0019】最初に、図4を参照して、ビートを検出する処理を詳細に説明する。

【0020】例えば、CD再生操作によりある曲の再生が開始される。そのとき、所定のスイッチ操作により図4によるビートの検出処理が開始される。

【0021】図4のステップ112で、低域側の音（上記ビート音）が入力されたか否かを判断し、ビート音が入力された場合には、ステップ114で、前回ビート検出時から所定時間U（例えば1.87 msec）が経過したか否かを判断する。ここで、所定時間U（1.87 msec）は、一般的なテンポのうち、やや早い方の数値である160（1分間に160拍）の曲の一拍の時間の半分の時間である。

【0022】前回ビート音検出時から所定時間U経過したと判断した場合には、ステップ116で、今回入力したビート音のレベルがしきい値を超えるか否かを判断する。上記3つの条件の1つでも満たさない場合には、ステップ112に戻る。よって、図5に示すように、サンプリング値S-X、S-Y等を対象外とすることができる。上記3の条件を満たす場合には、ステップ118で、上記3つの条件（ステップ112、114、116）を満たすビート音の入力個数を表わす変数Cが0か否かを判断する。変数Cが0の場合には、ステップ110、120で、変数1をインクリメントし、ステップ124で、領域Cに今回の音レベルを記憶して、ステップ

112に戻る。

【0023】一方、変数が0でない場合には、上記3条件を満たすビート音が複数入力された場合であり、この場合には、ステップ126で、領域Cに記憶されているレベルより、今回入力したビート音のレベルが、一定量大きいか否かを判断する。一定量大きくない場合には、今回入力したビート音を無視して、ステップ112に戻る。一方、一定量大きい場合には、ステップ128で、変数CがC0（例えば4）より大きいか否かを判断する。変数CがC0以下の場合にはステップ130で、変数Cを1インクリメントし、ステップ132で、領域Cに今回の音レベルを記憶して、ステップ112に戻る。一方変数CがC0より大きいと判断された場合には、上記3条件（ステップ112、114、116）を満足する音がC0個あった、即ち、図5に示すように、サンプリング値S1～S4があったと判断できる。すなわちビート音の立ち上がりを検出したと判断することができる。ステップ134で今回検出された音をビートとして設定し、ステップ136で、各領域Cのデータをリセットし、ステップ138で変数Cを初期化する。

【0024】なお、ビートの検出処理は、上記方法に限定されず、次のようにビートを検出するようにしてもよい。即ち、所定時間毎にビート音をサンプリングし、今回サンプリングしたビート音が前回ビートとして検出したときから上記所定時間U経過しているか否かを判断し、上記所定時間U経過していない場合には、今回のサンプリング値を保持する。一方、今回サンプリングしたビート音が前回ビートとして検出したときから上記所定時間U経過している場合には、今回のサンプリング値よりも所定個数、例えば、4個まえの全てのサンプリング値を取り込み、今回のサンプリング値が、取り込んだ過去所定個数（例えば、4個）の何れかのサンプリング値よりも所定レベル大きいか否かを判断する。今回のサンプリング値が、取り込んだ過去所定個数（例えば、4個）の全てのサンプリング値よりも所定レベル大きくな場合には、今回のサンプリング値を保持する。一方、今回のサンプリング値が、取り込んだ過去所定個数（例えば、4個）の何れかのサンプリング値よりも大きい場合には、ビートとして設定する。なお、上記では保持するサンプリング値は4個としている。ビートが検出されると、それを基にテンポ算出を実行する。

【0025】次に、図6を参照して、テンポを算出する処理を詳細に説明する。

【0026】図6のステップ142で、ビート時間を検出する。即ち、前回ビートが検出された時から今回ビートが検出された時までの時間（ビート時間）を検出する。なお、本実施の形態では、図7に示すように、今回のビート時間BT8を含めた過去8個のビート時間（BT1～BT8）を記憶する。

【0027】ステップ144で、過去のM個（例えば8

10

20

30

30

40

40

50

個）のビート時間（BT1～BT8）を取り込み、ステップ146で、今回のビート時間に近い（即ち、略同じ値）の過去のビート時間の個数Kを検出する。即ち、過去のM個のビート時間BT1～BT8の中に今回のビート時間BT8の±11msのビート時間の個数Kを検出する。

【0028】ステップ148で、個数KがN（例えば4（半数）等）以上か否かを判断する。即ち、今回のビート時間に近い過去のビート時間の個数がN以上か否かを判断する。

【0029】今回のビート時間に近い過去のビート時間の個数がN以上の場合には、ステップ150で、平均ビート時間を算出する。即ち、今回のビート時間と今回のビート時間に近い過去のビート時間との平均ビート時間を算出する。なお、今回のビート時間に近い過去のビート時間の個数がN未満の場合には、ステップ152に進む。

【0030】ところで、音源からの音楽のテンポは一定とは限らず、図8（A）や図8（B）に示すように、過去のビート時間の比が1対3である曲や3対1である曲があるので、ステップ152でこれを判定する。ステップ152で、過去のビート時間の比が1対3または3対1の場合には、ステップ154で、一拍の時間を平均ビート時間として再設定する。即ち、（1.5拍時間+0.5拍時間）/2により求める。

【0031】ステップ156で、今回テンポが算出できたか否かを判断する。即ち、今回のビート時間に近い過去のビート時間の個数KがN未満か否かを判断する。今回テンポが算出できた場合には、ステップ158で、

図9に示すように、表示部30の所定表示個所90H、90Lを点灯し、今回テンポが算出できなかった場合には、ステップ160で、表示部30の所定表示個所90H、90Lを点滅する。

【0032】ここで、ステップ158で点灯した表示部30は、一度テンポ算出がなされた時点から同一曲演奏中は続けて点灯するようにしてもよい。また、ステップ160で表示部30を点滅させるが、ある曲の演奏でテンポが算出できずに、曲が終了した場合に、点滅するようにしてもよい。こうすることで、曲の途中で表示部30が点灯、点滅を繰り返す動作を防止できる。

【0033】次のステップ162で、テンポとして、60を平均ビート時間で割った値を算出する。そして、ステップ164で、前に記憶していたビート時間に換えて今回のビート時間を記憶する。

【0034】以上の処理により、テンポを算出することができる。そして、本処理により算出されたテンポを用いて前述したリズムパターンの1小節時間Tを定める。よって、リズムパターンの1小節時間を、音楽の実際にテンポに変更（近づける）することができる。

【0035】また、テンポを算出する際、今回のビート

時間に近い過去のビート時間の個数がN個以上の場合に、今回のビート時間と今回のビートに近い過去のビート時間との平均ビート時間を算出して、テンポを算出する、即ち、細かいパターン（種々のリズム又は音楽ジャンルに対応する）認識せずにテンポを算出するので、テンポ算出時間を短縮することができる。

【0036】なお、テンポの算出処理は上記に限定されるものでなく、次のようにテンポを算出してもよい。即ち、上記のようにビート時間をM個検出する毎に、M個のビート時間の中の何れかのビート時間、例えば、最も古いビート時間と他のビート時間とを比較し、最も古いビート時間に近いビート時間の個数がN個ある場合、最も古いビート時間に近いN個のビート時間から上記のように平均ビート時間を算出して、上記のようにテンポを算出してもよい。

【0037】ところで、音声出力ICは、マイコンによって命令されてから、リズムパターンを繰り返し出力するが、リズムパターンのスタート時と音楽のビート発生時とがずれる場合がある。図10(A)には、曲のテンポに沿って周期的に発生するビート音（低域側の音）の発生の様子が示され、図10(B)には、音声出力ICによりリズムパターンが出力された様子が示されている。図10(A)及び図10(B)に示す例では、最初のリズムパターンP1は、そのスタート時はビート音B1と同期しており、次の1リズムパターンP2のスタート時もビート音B2と同期しているが、次の1リズムパターンP3のスタート時とビート音B3の出力タイミングが同期していない。このように、リズムパターンのスタート時と音楽のビート発生時とがずれているとリズムパターンが曲と合っていないように聞こえる。よって効果音としてのリズムパターンが耳障りなものとなっている。よって、リズムパターンのスタート時を、ビート音に同期させる必要がある。そこで、本実施の形態では、図11に示すリズムパターン同期処理ルーチンを実行する。

【0038】図11のステップ52で、リズムパターンを出力し始めた時から、算出されたテンポにより定まる1小節時間Tから所定時間（例えば80ms·e·c）tを引いた時間T-t経過したか否かを判断する。リズムパターンを出力し始めた時から、時間T-t経過した場合には、ステップ54でリズムパターンを出力し始めた時から、1小節時間T経過したか否かを判断する。ステップ54でリズムパターンを出力し始めた時から、1小節時間T経過していないと判断した場合には、ステップ56で、低域側の音（ビート音）が入力された否かを判断する。ビート音が入力された場合には、ステップ58以降の処理を実行し、ビート音が入力されていない場合には、ステップ54に戻る。ステップ54でリズムパターンを出力し始めた時から1小節時間T経過したと判断した場合には、ステップ72で1小節時間Tに所定時間t

を加算した時間T+t経過したか否かを判断する。ステップ72で時間T+t経過していないと判断した場合には、ステップ74で、低域側の音（ビート音）が入力されたか否かを判断する。ビート音が入力された場合には、ステップ76以降の処理を実行し、ビート音が入力されていない場合には、ステップ72に戻る。このようにステップ52、54、56、72、74は、リズムパターンのスタート時の±所定時間t内にビート音が入力されたか否かを監視している。

【0039】ステップ56でビート音が入力されたと判断した場合は、一つのリズムパターンの出力が完了する前に曲の次の1小節が始まる、即ち、リズムパターンの時間が長い（算出したテンポにより定まる1小節時間が長い）。つまり、算出したテンポが遅いと判断することができる。この場合、ステップ58でビート音が入力された時から、1小節時間T経過するまでの時間△T（誤差（ずれ量））を算出する。

【0040】誤差△Tがしきい値△T0より大きいか否かを判断する。誤差△Tがしきい値△T0より大きい場合には、算出したテンポが合っていない場合（許容範囲外）であり、ステップ66に進む。一方、誤差△Tがしきい値△T0以下の場合には、テンポが合っている（許容範囲内）と判断でき、ステップ54に戻る。

【0041】ステップ66で、誤差振分時間を算出する。誤差振分時間は、誤差△Tを所定倍、例えば、2倍した時間を、リズムパターン内の所定分割数（16または12）で割った時間である。そして、ステップ68で、リズムパターンの各分割時間に、誤差振分時間を振り分ける。即ち、算出したテンポが遅いので、各分割時間を誤差振分時間分短くする。

【0042】ここで、誤差振分時間として誤差△Tを所定倍（例えば、2倍）しているのは次の通りである。即ち、単に誤差△Tから誤差振分時間を求めて上記のように振分けると、マイコンの処理時間の遅れにより、この遅れ時間が繰り返されてテンポを正確に実際の値にすることはできない。一方、誤差△Tを所定倍でし誤差振分時間を求めて上記のように振分けると、マイコンの処理時間の遅れがあっても、テンポを正確に実際の値にすることができます。そこで、誤差振分時間として誤差△Tを所定倍（例えば、2倍）している。なお、多数の実験から誤差△Tを2倍にすることがテンポを正確に実際の値にすることことができたことを発見した。

【0043】一方ステップ74でビート音が入力された場合は、図10に示すように、一つのリズムパターンの出力が完了した後（即ち、次のリズムパターンのスタート後）にビート音が検出されたと判断することができる。即ち、リズムパターンの時間が短い（算出したテンポにより定まる1小節時間が短い）。つまり、算出したテンポが速いと判断することができる。

【0044】この場合には、ステップ76で1小節時間

T経過時から現在（ビート音入力時）までの時間△T（誤差（ずれ量））を算出する。

【0045】次のステップ84で、前述した振分時間を算出し、ステップ86でリズムパターンの各アクセント間隔に誤差振分時間を振り分ける。即ち、リズムパターンの各アクセント時間と誤差振分時間分長くする。

【0046】ここで、前述するように、テンポを算出して1小節時間Tを定めるようにしているが、本発明はこれに限定されるものではなく、上記時間△Tより1小節時間Tを修正してもよい。

【0047】即ち、図10に示す例を更に詳細に説明すると、例えば、リズムパターンP3のスタート時よりビート音B3が30 msec遅かったとすると、リズムパターンが速かったと判断し、その1リズムパターンP3を出力する期間（シーケンス）を60 msec遅らせる。この場合、ユーザーによって選択されているリズムパターンが、図3（A）、図3（B）のような場合、 $60 \div 16 = 3.75 \text{ msec}$ を各アクセント間隔にプラスする。即ち、アクセント間隔にプラスする。なお、本実施の形態では、1 msec以下はカウントしないため、余りの時間を切り捨てる。

【0048】以上のように、ほぼ規則的にビートが出ているものは、所定時間（例えば、±80 msec）以内で補正され、リズムパターンと音楽との同期がとれる。

【0049】なお、テンポ値を81～160としており、81の半分のテンポから160の倍のテンポまで観測することができるようになっているため、最小ビート間隔は187 msecと仮定している。よって監視時間は±80 msecという固定値にしている。

【0050】以上の処理により、図12（A）に示すように音楽信号から、図12（B）に示すようにビート音が検出され、このビート音から算出されるテンポに従って1リズムパターンを図12（C）、図12（D）に示すように出力する。図12（C）に示すように、最初のスタート時は同期していても、次のリズムパターンのスタート時がビート音とずれるような場合があつても、前述したように、リズムパターンのスタート時の前後所定時間（±80 msec）を監視し、リズムパターンのスタート時とビート音検出時との誤差を、そのリズムパターン内で相殺しているため、同期させることができる。

【0051】ところで、図12（D）に示すように、リズムパターンの1小節時間と音楽の1小節時間とが一致していても、リズムパターンのスタートタイミングがずれる場合がある。この場合には、±80 msec以内になるべく入るように、テンポが算出された瞬間近くのビートとリズムパターンのスタート時を同期させてスタートするようにする。

【0052】次に、図13を参照して、操作部の所定の操作によりスタートする効果音付加処理ルーチンを説明

する。

【0053】図13のステップ92で、低域側（100 Hz～300 Hz）の第1の周波数（例えば105 Hz）の第1の音（ビート音）が入力されたか否かを判断する。ビート音が入力されていない場合には、ステップ94で高域側（3 kHz～15 kHz）内の第2の周波数（例えば3.4 kHz）の第2の音（アタック音）が入力されたか否かを判断する。アタック音が入力されていない場合には、ステップ92に戻る。

10 【0054】ビート音が入力されたと判断した場合には、ステップ96で、ビート音に対応して定められた第1の効果音を出力するように音声出力IC32を制御し、ステップ92で、第1の音（ビート音）に対応して定められた表示箇所90Lを点滅させて、ステップ94に進む。

【0055】一方、アタック音が入力された場合には、ステップ100で、アタック音に対応して定められた第2の効果音を出力するように音声出力IC32を制御し、ステップ102で、アタック音に対応して定められた表示箇所90Hを点滅させて、ステップ92に進む。

20 【0056】このように、ビート音やアタック音が検出される毎にこれらの音に対応して定められた効果音を出力すると共にこれらの音に対応して定められた表示箇所を点滅させるので、視覚的に効果を発生させることができる。また、視覚的にもテンポを取ることができる。

【0057】以上はビート音を用いて、テンポを算出したりリズムパターンのスタート時を同期させたりしているが、本発明はこれに限定されるものではなく、アタック音を用いて、テンポを算出したりリズムパターンのス

30 タート時を同期させたりしてもよく、更に、ビート音ばかりでなくアタック音を用いて、テンポを算出したりリズムパターンのスタート時を同期させたりし、ビート音を用いて、テンポを算出したりリズムパターンのスタート時を同期させたりすることができない場合に、アタック音を用いて、テンポを算出したりリズムパターンのスタート時を同期させたりしてもよい。

【0058】【発明の効果】以上説明したように本発明は、連続する2つの周期音時間を連続して求め、今回求めた周期音時間と過去求めたM個の周期音時間の中の今回求めた周期音時間に近い周期音時間とから平均周期音時間を求めてテンポを算出する、即ち、複数のパターンをマッチング処理しないで、テンポを算出するので、簡易にテンポを算出することができ、よって、テンポ算出する時間を短くすることができる、という効果を有する。

【図面の簡単な説明】

【図1】本実施の形態に係る音響装置の外観図である。

【図2】本実施の形態に係る音響装置の主制御系を示したブロック図である。

50 【図3】複数のリズムパターンの表を示した図である。

【図4】ビート検出処理ルーチンを示したフローチャートである。

【図5】ビートの検出の様子を示す図である。

【図6】テンポ算出処理ルーチンを示したフローチャートである。

【図7】テンポが一定な曲のビート音の発生タイミングを示した図である。

【図8】テンポが1対3または3対1の曲のビートの発生の様子を示した図である。

【図9】表示部の詳細図である。

【図10】ビート音とリズムパターンのタイミングチャートである。

【図11】リズムパターン同期処理ルーチンを示したフローチャートである。

【図12】ビート音とリズムパターンのずれを示したタイミングチャートである。

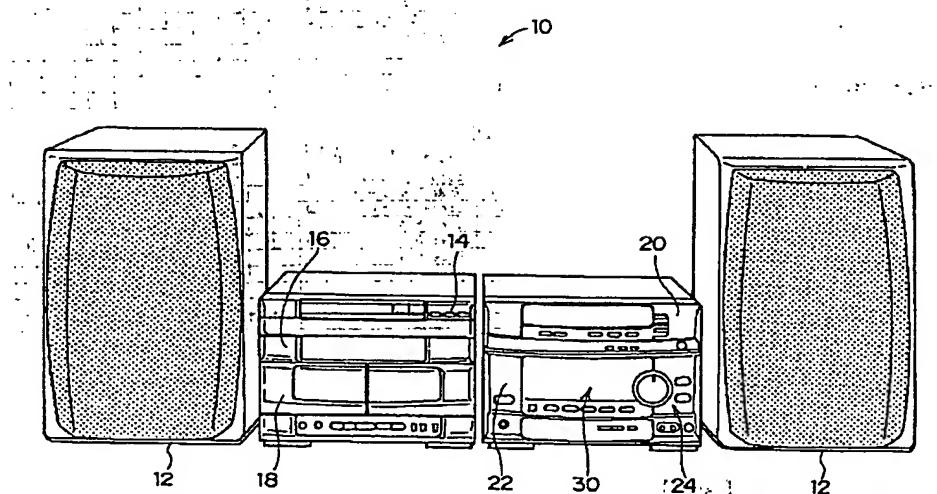
【図13】効果音付加処理ルーチンを示した図である。

【図14】従来の音響装置の制御系の一部を示したブロック図である。

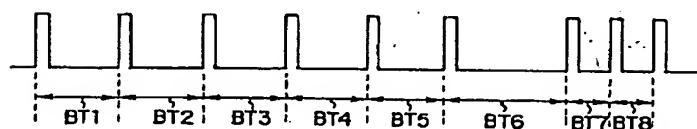
【符号の説明】

10	オーディオシステム
12	スピーカ
14	ミニディスクレコーダ
16	CD再生装置
18	カセットテープレコーダ
20	チューナ
22	アンプ
24	操作部
26	マイコン
28	スペクトラムアナライザIC
30	表示部
32	音声出力IC

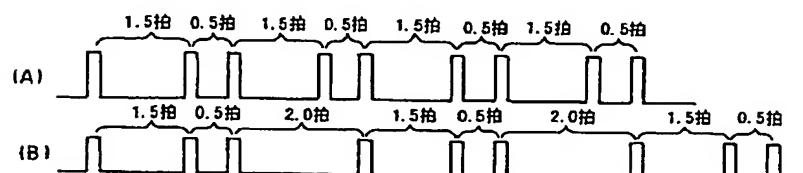
【図1】



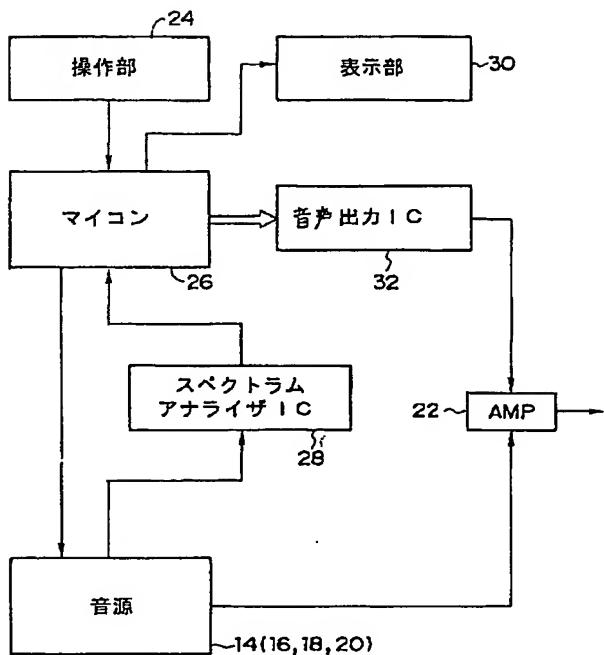
【図7】



【図8】



【図2】



【図3】

(A) Rhythm Title: HARD ROCK 1
Tempo = J 100 (msec/Beat)

No	音源名\ビート	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	Snare Drum 1			①		①		①		①		①		①			
2	Bass Drum	①		①	①	①	①	①	①	①	①	①	①	①	①	①	
3	Ride Cymbal	②	②	②	②	②	②	②	②	②	②	②	②	②	②	②	
4																	
5																	
6																	
7																	
8																	
9																	
10																	
11																	
12																	
13																	
14																	
15																	
16																	

(B) Rhythm Title: HARD ROCK 2
Tempo = J 100 (msec/Beat)

No	音源名\ビート	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	Open Hi Hat			②				②			②						
2	Snare Drum 1			①				①			①						
3	Bass Drum	①		①	①	①	①	①	①	①	①	①	①	①	①	①	
4	Crash Cymbal	②															
5	High Tom												①				
6	Mid Tom											①					
7	Low Tom												①				
8																	
9																	
10																	
11																	
12																	
13																	
14																	
15																	
16																	

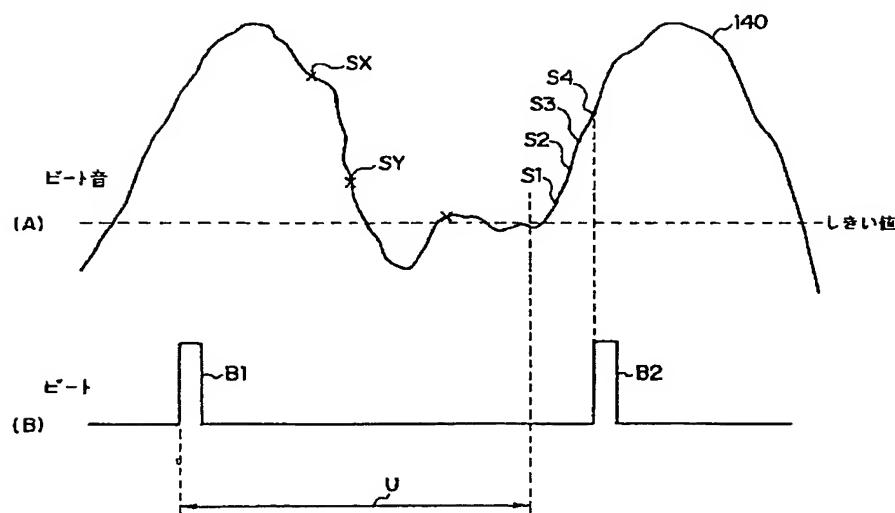
(C) Rhythm Title: WALTZ 1
Tempo = J 120 (msec/Beat)

No	音源名\ビート	1	2	3	4	5	6	7	8	9	10	11	12
1	Close Hi Hat	②	②	①	①	①	①	①	①	①	①	①	①
2	Bass Drum	①											
3	Ride Cymbal	②											
4	Rim Shot		①					①					
5													
6													
7													
8													
9													
10													
11													
12													

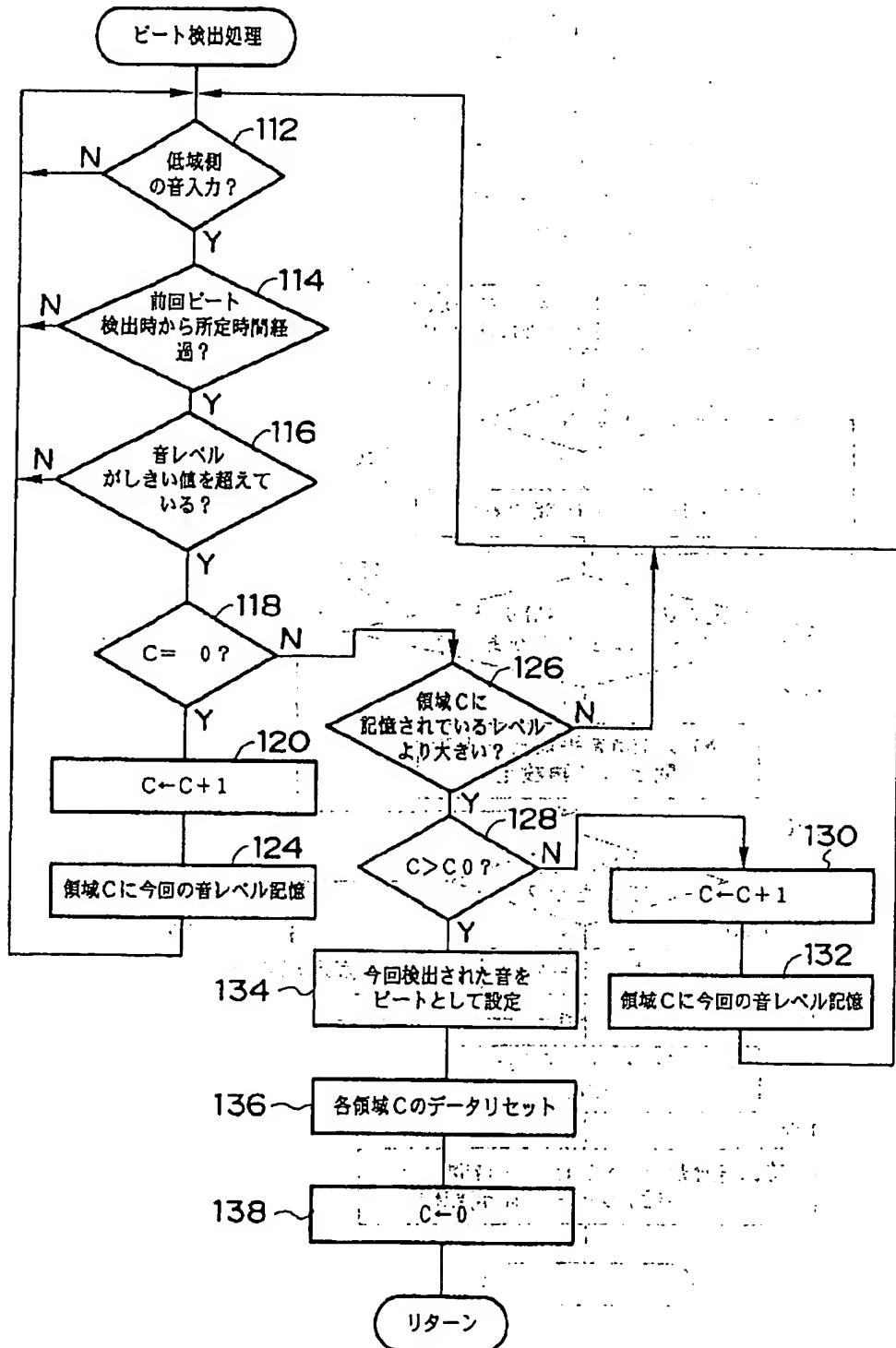
(D) Rhythm Title: WALTZ 2
Tempo = J 120 (msec/Beat)

No	音源名\ビート	1	2	3	4	5	6	7	8	9	10	11	12
1	Close Hi Hat	②	②	②	②	②	②	②	②	②	②	②	②
2	Open Hi Hat	①											
3	Bass Drum	①											
4	Ride Cymbal							②					
5													
6													
7													
8													
9													
10													
11													
12													

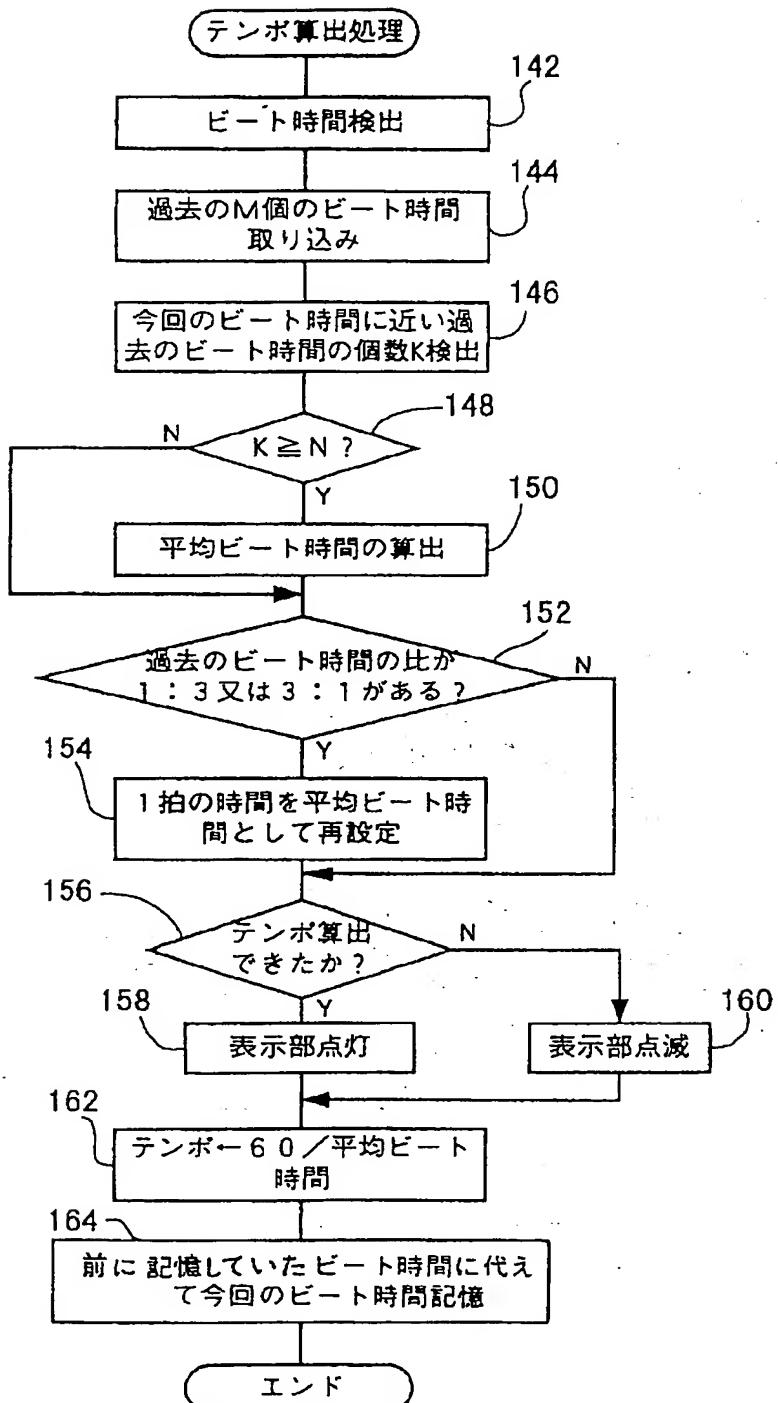
【図5】



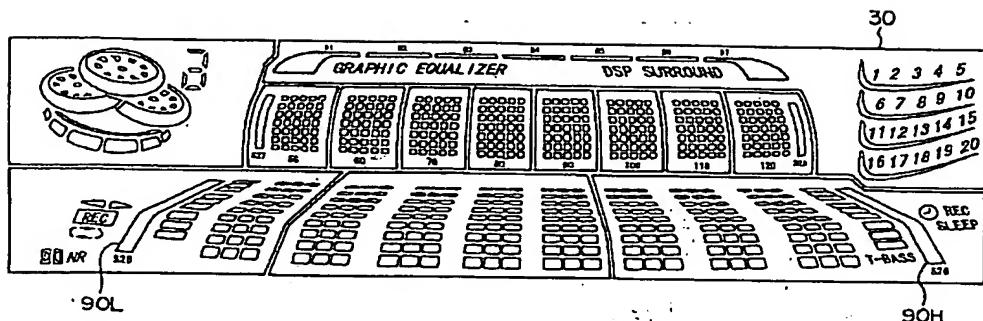
【図4】



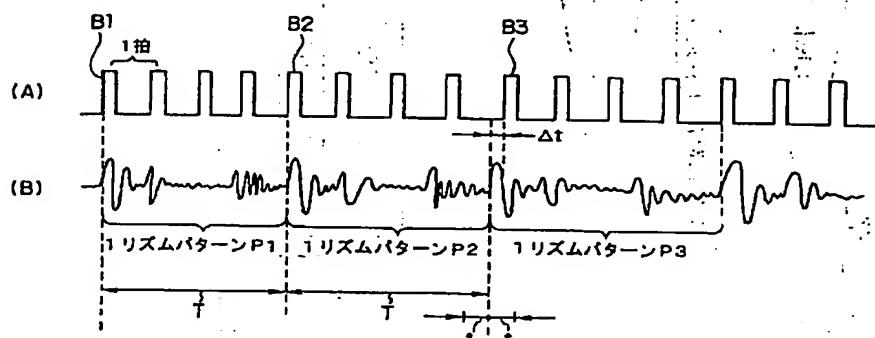
【図6】



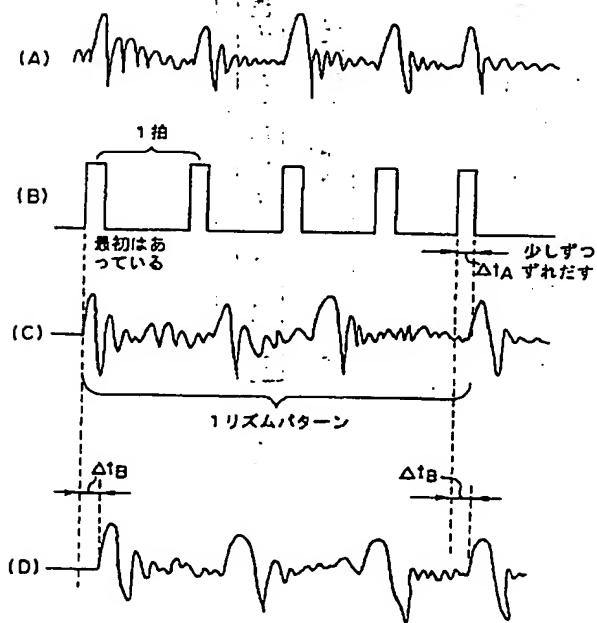
【図 9】



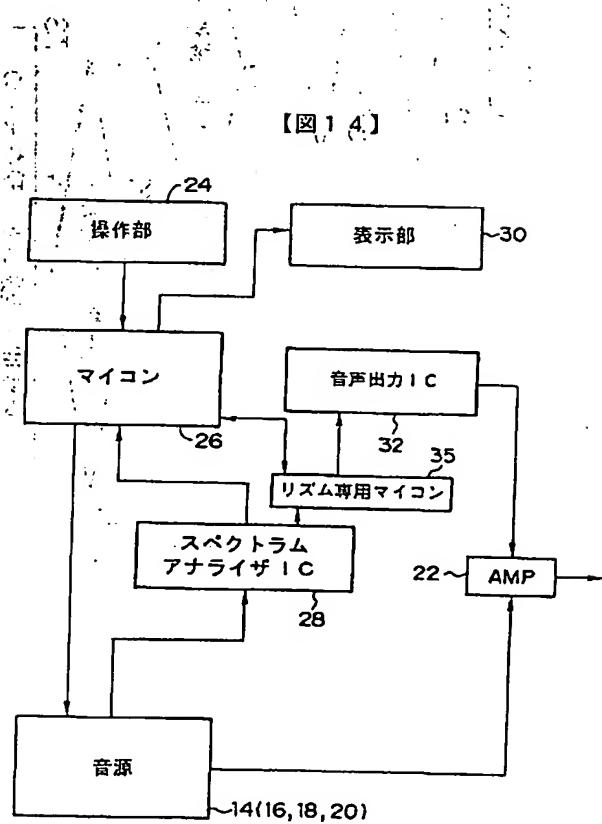
【図 10】



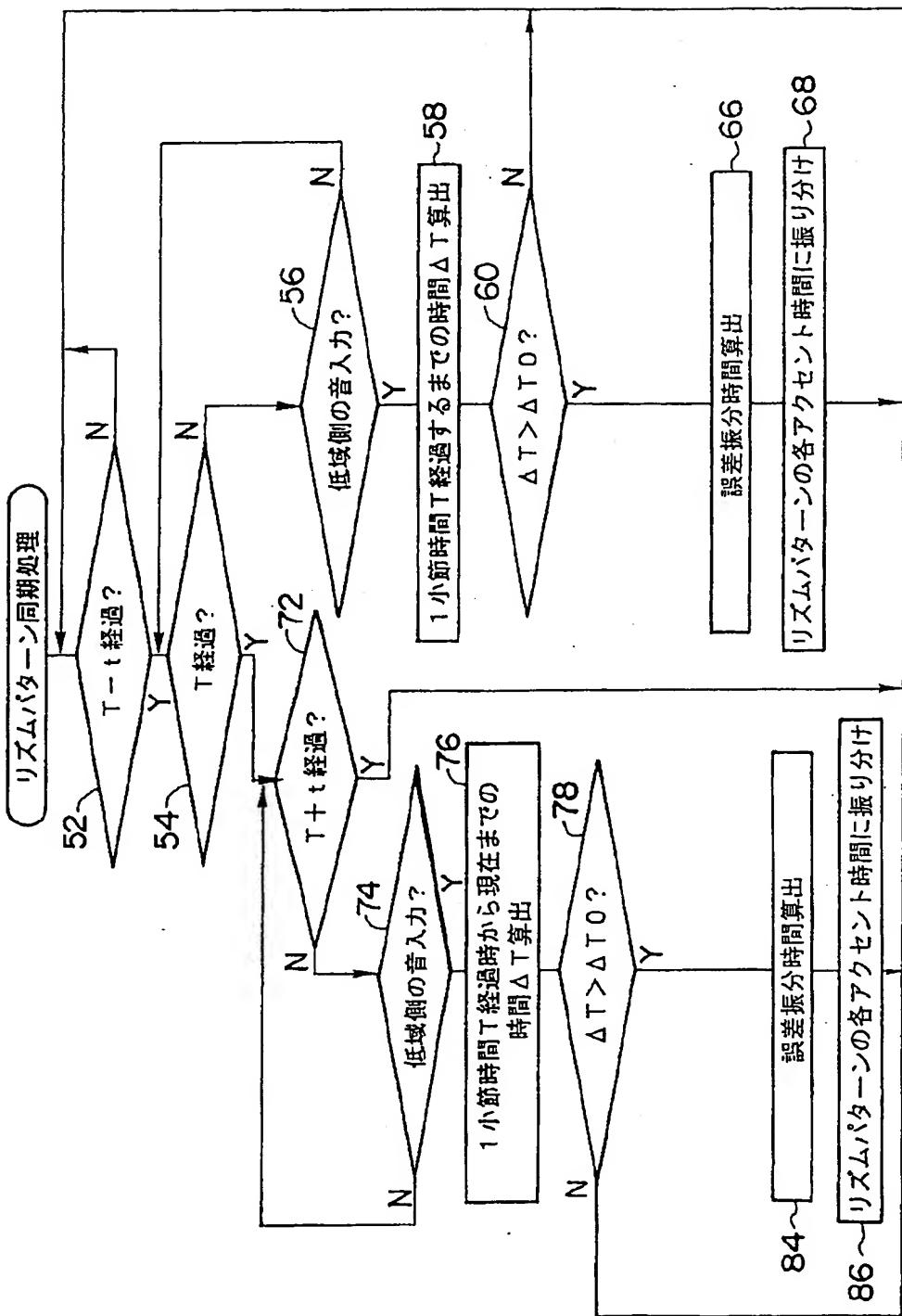
【図 12】



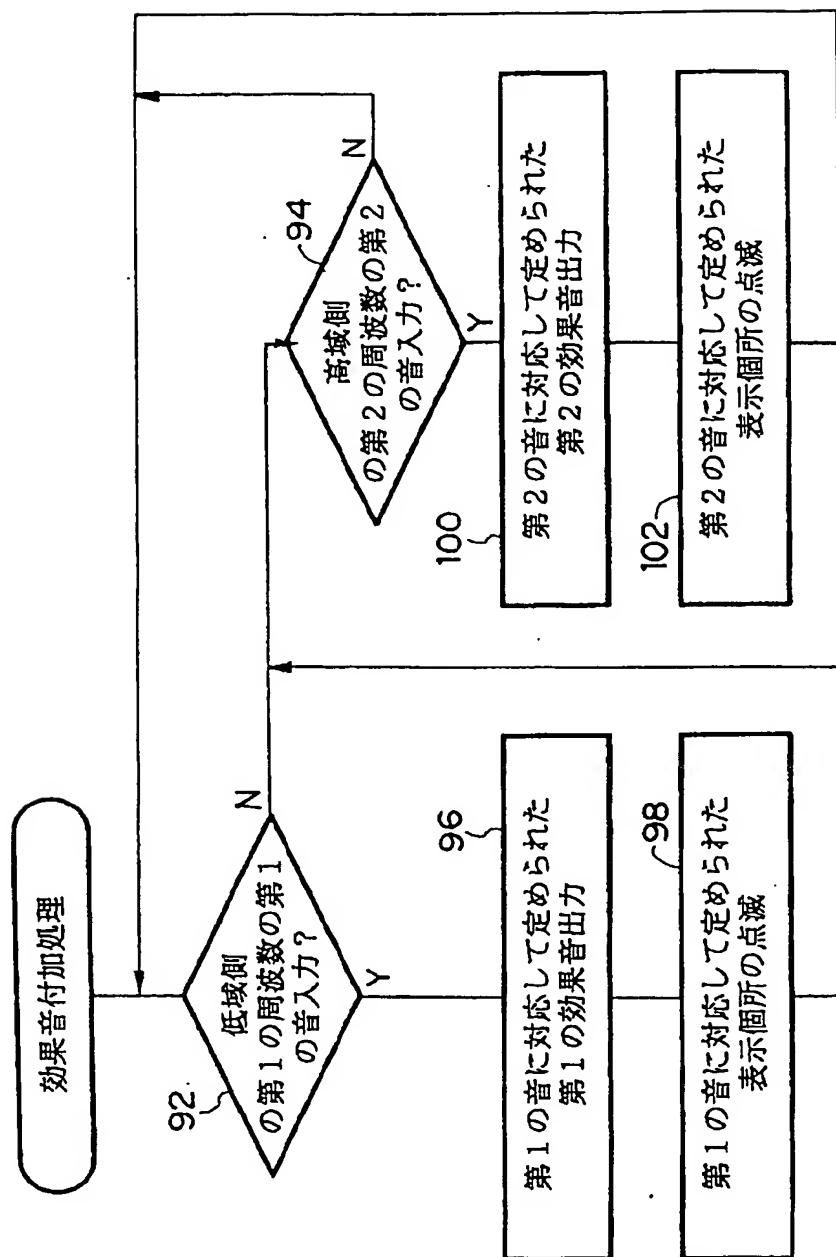
【図 14】



【図 11】



【図13】



フロントページの続き

(51) Int. Cl. 7

H 04 R 3/00

識別記号

310

F 1

H 04 R 3/00

テマコード(参考)

310

THIS PAGE BLANK (USPTO)